After finishing my Node-Red dashboard app, and boy, it's awesome. Everything works great, but I wanted to bind it to port 80 on my nginx server.

A more popular approach is to set up Nginx as a reverse proxy by having it bind to the desired port, forwarding all incoming traffic to my node red dashboard.

Nginx is a high performance, open source web server (similar to Apache) that is widely-used.

The main benefit of Nginx is the fact that it takes care of transport optimization.

**Installing Nginx**

sudo apt-get install nginx

**Configuring Nginx**

Next, we'll need to configure Nginx so that it forwards traffic to my dashboard. Let's start off by removing the default configuration file:

sudo rm /etc/nginx/sites-enabled/default

Next, create a new file in /etc/nginx/sites-available/ called node and open it with nano:

sudo nano /etc/nginx/sites-available/node

Paste the following code in the file and make sure to change [example.com](http://example.com/) to your domain (or IP), and 1880 to your Node-Red Dashboard port:

server {  
listen 80;  
server\_name [example.com](http://example.com/);  
    location /ui {  
    proxy\_set\_header X-Forwarded-For $remote\_addr;  
    proxy\_set\_header Host $http\_host;  
    proxy\_pass "[http://127.0.0.1:1880](http://127.0.0.1:1880/)";  
    }  
}

The proxy\_pass declaration configures Nginx to act as a reverse proxy by forwarding all incoming requests on port 80 to Node-Red dashboard on port 1880, on behalf of the client.

Next, we need to symlink our configuration to sites-enabled for it to be used by Nginx, since it's currently in sites-available:

sudo ln -s /etc/nginx/sites-available/node /etc/nginx/sites-enabled/node

Applying the Configuration

Let's restart Nginx so that it loads our configuration:

sudo service nginx restart

run

sudo service nginx status

and you should see something like:

\* nginx.service - A high performance web server and a reverse proxy server  
Loaded: loaded (/lib/systemd/system/nginx.service; enabled)  
Active: active (running) since Sat 2017-02-11 14:14:14 UTC; 306ms ago  
Process: 3103 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid (code=exited, status=0/SUCCESS)  
Process: 3111 ExecStart=/usr/sbin/nginx -g daemon on; master\_process on; (code=exited, status=0/SUCCESS)  
Process: 3108 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master\_process on; (code=exited, status=0/SUCCESS)  
Main PID: 3113 (nginx)  
CGroup: /system.slice/nginx.service  
|-3113 nginx: master process /usr/sbin/nginx -g daemon on; master\_process on;  
|-3114 nginx: worker process  
|-3115 nginx: worker process  
|-3116 nginx: worker process  
`-3119 nginx: worker process  
  
Feb 11 14:14:14 alexapi nginx[3108]: nginx: [warn] server name "127.0.0.1:1880/ui" has suspicious symbols in /etc/nginx/sites-enabled/node:3  
Feb 11 14:14:14 alexapi nginx[3111]: nginx: [warn] server name "127.0.0.1:1880/ui" has suspicious symbols in /etc/nginx/sites-enabled/node:3  
Feb 11 14:14:14 alexapi systemd[1]: Started A high performance web server and a reverse proxy server.

All set! Nginx will now forward all incoming requests to your app and even survive a server crash, since it automatically starts up with your machine.